

# A Quantum Algorithm for Finding Common Matches Between Databases with Reliable Behavior

Khaled El-Wazan <sup>\*1</sup>

<sup>1</sup>Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Egypt

## Abstract

Given two or more databases each of unstructured  $2^n$  entries, we propose a quantum algorithm to find the common entries between those databases. The proposed algorithm requires  $\mathcal{O}(\sqrt{2^n})$  queries to find the common entries. The proposed algorithm constructs an oracle to mark common entries, and then uses a variation of amplitude amplification technique with reliable behavior to increase the success probability of finding them.

## 1 Introduction

Given  $\kappa$  databases with  $2^n$  unstructured entries, it is required to find the joint entries between those databases. Considering this problem in classical computers, an intuitive approach is to count the entries from those databases and store them in a memory which keeps track of each entry and its number of occurrences, and then iterate over that memory and observe when the number of occurrences of certain entries equal to  $\kappa$ . This procedure requires at most  $\mathcal{O}(\kappa 2^n)$  queries.

Quantum computers are inherently probabilistic devices which promise to significantly accelerate certain types of computations compared to classical computers, by utilizing quantum phenomena like entanglement and superposition.

One of the early algorithms, which demonstrated the keynote of quantum parallelism and quantum interference to get some global property of a given function, is Deutsch-Jouza algorithm [1]. This algorithm solves this problem: given a black-box reversible Boolean function  $f$  which maps  $n$  inputs to either 0 or 1, check whether the function  $f$  is a balanced Boolean function or a constant Boolean function. On one hand, solving this problem classically will require

---

<sup>\*</sup>khaled\_elwazan@alex-sci.edu.eg

$\mathcal{O}(2^n)$  oracle calls to test for this property, on the other hand, using a quantum computer, it will require only a single oracle call, *i.e.*  $\mathcal{O}(1)$  using a uniform superposition of all possible input to be queried to the function.

In 1998, Burhman *et al.* introduced an algorithm [2] similar to the common entries problem: given two remotely separated schedules of unknown free slots out of  $2^n$  slots, find a common slot between those two schedules, in as minimum communication bits sent as possible. This algorithm required  $\mathcal{O}(\sqrt{2^n} \log_2 2^n)$  communication complexity and  $\mathcal{O}(k\sqrt{2^n})$  query calls, with  $k$  trails and error at most  $2^{-k}$ . Later in 2002, L. Grover proposed an algorithm [3] to solve that problem with  $\mathcal{O}(\sqrt{\epsilon 2^n} \log_2 2^n)$  computation complexity.

In 2008, Pang *et al.* introduced a quantum algorithm for set operations [4]. In that literature, Pang *et al.* provided a subroutine to find a common intersected element between two sets of size  $2^n$  and  $2^m$  elements in  $\mathcal{O}(\sqrt{2^{m+n}})$ , using a similar algorithm proposed in [5].

In 2012, A. Tulsi proposed a quantum algorithm [6] to find a single common element between two sets in  $\mathcal{O}(\sqrt{2^n})$  using an ancilla qubit to mark the common solution with phase-shift and applying amplification algorithm to increase the success probability of the desired result.

The aim of this paper is to propose an algorithm to find common matches  $M$  between given  $\kappa$  databases of  $2^n$  elements each. Each given database uses a black-box oracle with  $n$  input variables to identify its element. The proposed algorithm can find a match among the common entries using a new oracle  $U_{\tilde{h}}$  which is constructed from the set of all given black-box oracles  $\tilde{h}$ . The new oracle  $U_{\tilde{h}}$ , is then used along with amplitude amplification technique based on partial diffusion operator, to increase the success probability of finding the desired results. As well, the algorithm works with probability of success at least  $2/3$ .

The paper is organized as follows: sect. 2 defines the problem of multiple common solutions problem. Section 3 depicts an amplitude amplification algorithm with reliable behavior. Section 4 introduces a new oracle  $U_{\tilde{h}}$  which will be used to identify the common solutions. Section 5 presents the proposed algorithm. Section 6 will analyze the proposed algorithm and compare it to other literature, followed by a conclusion in sect. 7.

## 2 Common Matches Problem

Consider having a set of lists  $\tilde{h}$  of size  $\kappa \geq 2$ . Each list  $L_i$  in  $\tilde{h}$  is of  $N = 2^n$  unstructured entries, which has an oracle  $U_i$  that is being used to access those entries in  $L_i$ . Each entry  $j \in L_i = \{0, 1, \dots, N - 1\}$  in the list  $L_i$  is mapped to either 0 or 1 according to any certain property satisfied by  $j$  in  $L_i$ , *i.e.*  $f_i : L_i \rightarrow \{0, 1\}$ . The common elements problem is stated as follows: find the entry  $j \in L_i$  such that  $\forall L_i \in \tilde{h}, f_i(j) = 1$ .

### 3 Amplitude Amplification

Consider having a list  $L$  of  $N = 2^n$  of unstructured entries, which has an oracle  $U_f$  that is being used to access those entries. Each entry  $i \in L = \{0, 1, \dots, N-1\}$  in the list  $L$  is mapped to either 0 or 1 according to any certain property satisfied by  $i$  in  $L$ , *i.e.*  $f : L \rightarrow \{0, 1\}$ . The amplitude amplification problem is stated as follows: find the entry  $i \in L$  such that  $f(i) = 1$ .

In 1996, L. Grover proposed a unique approach to solve this typical problem with quadratic speed-up compared to classical algorithms [7]. The algorithm Grover proposed takes advantage of quantum parallelism to solve this problem by preparing a perfect superposition of all the possible  $N$  entries corresponding to the list  $L$ , after that it starts marking the solution using phase shift of  $-1$  using the oracle  $U_f$ , followed by amplifying the amplitude of the solution using inversion about the mean operator. It was shown in [7, 8] that the algorithm requires  $\pi/4\sqrt{N}$  iteration to optimally find a solution to the search problem with high probability, assuming there is only one solution  $j \in L$  that satisfies the oracle  $U_f$ .

Boyer *et al.* later generalized Grover's quantum search algorithm to fit the purpose of finding multiple solutions  $M$  to the oracle  $U_f$ , *i.e.*  $\forall p$ , for which  $1 \leq p \leq M \leq 3N/4$ ,  $f(j_p) = 1$ , to require a number of  $\pi/4\sqrt{N/M}$  iterations of the algorithm [5]. For the case of unknown number of solutions  $M$  to the oracle, an algorithm [9] was proposed to find such number  $M$ . However, the generalized quantum search algorithm has shown to exponentially fail in the case of  $M > 3N/4$  [5, 8].

Younes *et al.* introduced a variation of the generalized quantum search algorithm [10] with reliable behavior in case of multiple solutions to the oracle  $U_f$ , *i.e.*  $1 \leq M \leq N$ , and for fewer number of matches  $M$ , Younes *et al.* algorithm runs in quadratic speed-up similar to Grover's algorithm.

In the case of known multiple solutions  $M$  for a list  $L$  of size  $N = 2^n$ , Younes *et al.* algorithm is outlined as follows:

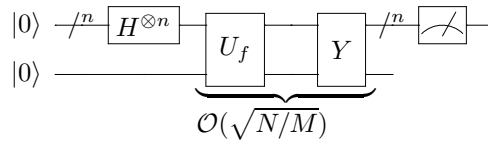


Figure 1: Quantum circuit for the quantum search algorithm [10].

1. Prepare a quantum register with  $n + 1$  qubits in a uniform superposition

$$|\varphi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |0\rangle. \quad (3.1)$$

2. Iterate the algorithm for  $\pi/(2\sqrt{2})\sqrt{N/M}$  times by applying the partial

diffusion operator  $Y$  on the state  $U_f|\varphi\rangle$  in each iteration, such that it performs the inversion about the mean on a subspace of the system, where

$$Y = (H^{\otimes n} \otimes I)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I). \quad (3.2)$$

At any iteration  $q \geq 2$ , the system can be described as follows [10]:

$$U_f|\varphi^q\rangle = a_q \sum_{i=0}^{N-1''} (|i\rangle \otimes |0\rangle) + c_q \sum_{i=0}^{N-1'} (|i\rangle \otimes |0\rangle) + b_q \sum_{i=0}^{N-1'} (|i\rangle \otimes |1\rangle). \quad (3.3)$$

where the amplitudes  $a_q$ ,  $b_q$  and  $c_q$  are recursively defined as follows:

$$a_q = 2\langle\alpha_q\rangle - \alpha_{q-1}, \quad b_q = 2\langle\alpha_q\rangle - c_{q-1}, \quad c_q = -b_{q-1}, \quad (3.4)$$

and

$$\langle\alpha_q\rangle = \left( \left(1 - \frac{M}{N}\right)a_{q-1} + \left(\frac{M}{N}\right)c_{q-1} \right). \quad (3.5)$$

For this algorithm, the success probability is as follows [10]:

$$P_s = (1 - \cos(\theta)) \left( \frac{\sin^2((q+1)\theta)}{\sin^2(\theta)} + \frac{\sin^2(q\theta)}{\sin^2(\theta)} \right), \quad (3.6)$$

where  $\cos(\theta) = 1 - M/N$ ,  $0 < \theta \leq \pi/2$ , and the required number of iterations  $q$  is given by:

$$q = \left\lfloor \frac{\pi}{2\theta} \right\rfloor \leq \frac{\pi}{2\sqrt{2}} \sqrt{\frac{N}{M}}. \quad (3.7)$$

Although Younes *et al.* variation of quantum amplitude amplification algorithm runs slower compared to Grover's algorithm by  $\sqrt{2}$  for small  $M/N$ , but this algorithm is more reliable with high probability than Grover's algorithm for multiple matches  $M$  [10].

## 4 Constructing the Oracle $U_{\tilde{h}}$

In this section, the given set of oracles  $\tilde{h}$  will be used to construct the oracle  $U_{\tilde{h}}$  which will be used for finding the common solutions  $M$  between the oracles in the set  $\tilde{h}$ . For the sake of simplification, we will provide a simple illustration for the oracle  $U_{\tilde{h}}$  assuming that the size of the set  $\tilde{h}$  is only  $\kappa = 2$  oracles, and after that we will propose the generalized form of the oracle  $U_{\tilde{h}}$  for multiple oracles  $\kappa \geq 2$ .

#### 4.1 The Oracle $U_h$ for 2 Functions

Given that  $\kappa = 2$  oracles  $U_A$  and  $U_B$  which map the elements of black-box functions  $f_A$  and  $f_B$  of  $n$  input qubits to either 0 or 1, it is required to find the common solutions  $M$  between them such that  $1 \leq M \leq N$ .

A quantum circuit for the oracle  $U_h$  can be constructed as follows:

$$U_h = (U_B \times I_\kappa)(U_A \times I_\kappa)(I_n \times U_\kappa)(U_B \times I_\kappa)(U_A \times I_\kappa), \quad (4.1)$$

where  $I_\kappa$  and  $I_n$  are the identity matrices of size  $\kappa = 2$  and  $n$  respectively, and the oracle  $U_\kappa$  represents the function  $f_\kappa(i)$ :

$$f_\kappa(i) = f_A(i) \cdot f_B(i), \quad (4.2)$$

where  $\cdot$  is the addition modulo 2.

An illustration of this circuit is shown in fig. 2.

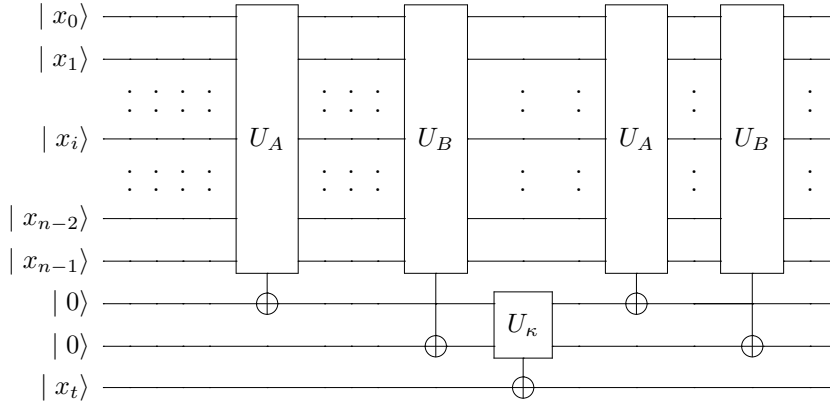


Figure 2: A quantum circuit for the proposed oracle  $U_h$  for  $\kappa = 2$  functions.

To clarify the effect of the proposed oracle  $U_h$ , let's analyze that effect on a uniform superposition as follows:

1. *Register Preparation.* Prepare a quantum register of size  $n + 3$  qubits in the state  $|0\rangle$ , where the last 3 qubits will be utilized as extra space to compute the oracles  $U_A$ ,  $U_B$  and the common solutions between them:

$$|\varphi_0\rangle = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes 3}. \quad (4.3)$$

2. *Register Initialization.* Apply Hadamard gates on the first  $n$  qubits to get a uniform superposition of all the possible  $N = 2^n$  states:

$$\begin{aligned}
|\varphi_1\rangle &= H^{\otimes n}|\varphi_0\rangle \\
&= H^{\otimes n}|0\rangle^{\otimes n} \otimes |0\rangle^{\otimes 3} \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |0\rangle^{\otimes 3}.
\end{aligned} \tag{4.4}$$

3. *Applying the Oracle  $U_A$ .* Apply the oracle  $U_A$  on the register to mark all possible solutions of the function  $f_A$  in the first extra qubit, where non-solutions will be marked with  $|0\rangle$  and the solutions will be marked with  $|1\rangle$ :

$$\begin{aligned}
|\varphi_2\rangle &= (U_A \times I_2)|\varphi_1\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |f_A(i)\rangle \otimes |0\rangle^{\otimes 2}.
\end{aligned} \tag{4.5}$$

4. *Applying the Oracle  $U_B$ .* Apply the oracle  $U_B$  on the register to mark all possible solutions of the function  $f_B$  in the second extra qubit, where the non-solutions will be marked with  $|0\rangle$  and the solutions will be marked with  $|1\rangle$ :

$$\begin{aligned}
|\varphi_3\rangle &= (U_B \times I_2)|\varphi_2\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |f_A(i)\rangle \otimes |f_B(i)\rangle \otimes |0\rangle.
\end{aligned} \tag{4.6}$$

5. *Applying the Oracle  $U_\kappa$ .* Apply the oracle  $U_\kappa$  on the register to mark all possible common solutions between the functions  $f_A$  and  $f_B$  in the third extra qubit, where non-common solutions will be marked with  $|0\rangle$  and the common solutions will be marked with  $|1\rangle$ :

$$\begin{aligned}
|\varphi_4\rangle &= (I_n \times U_\kappa)|\varphi_3\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |f_A(i)\rangle \otimes |f_B(i)\rangle \otimes |f_\kappa(i)\rangle,
\end{aligned} \tag{4.7}$$

where  $f_\kappa(i)$  is defined as in eq. (4.2).

6. *Applying  $U_B U_A$ .* Apply both the oracles  $U_B U_A$  to remove any entanglement between the solutions of both oracles from the first and the second extra qubits, and reset them to their initial state  $|0\rangle^{\otimes 2}$ :

$$\begin{aligned}
|\varphi_5\rangle &= (U_B \times I_2)(U_A \times I_2)|\varphi_4\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |0\rangle^{\otimes 2} \otimes |f_\kappa(i)\rangle.
\end{aligned} \tag{4.8}$$

Ignoring the extra qubits, this equation can be rewritten as follows:

$$|\varphi_5\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} ''(|i\rangle \otimes |0\rangle) + \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} '(|i\rangle \otimes |1\rangle), \tag{4.9}$$

where  $\sum''$  are all the possible uncommon solutions between the functions  $U_A$  and  $U_B$  marked with  $|0\rangle$ , and  $\sum'$  are all the possible common solutions between those functions marked with  $|1\rangle$ .

## 4.2 The Oracle $U_h$ for a Set of $\kappa$ Functions

Given that  $\kappa \geq 2$  oracles of  $n$  input qubits, we illustrate the circuit of the oracle  $U_h$  in fig. 3.

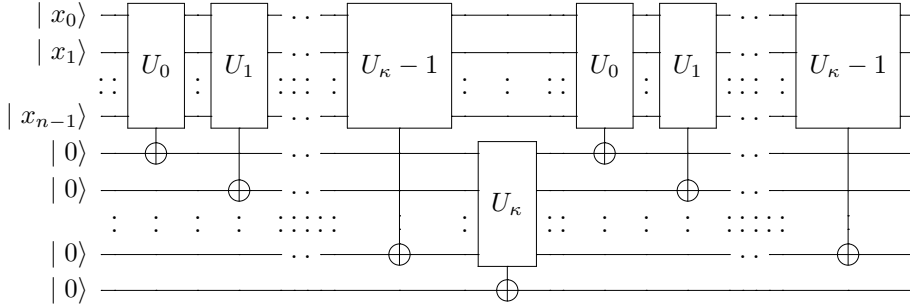


Figure 3: A quantum circuit for the proposed oracle  $U_h$  for  $\kappa$  functions.

The oracle  $U_h$  is generally defined as follows:

$$U_h = \prod_{j=0}^{\kappa-1} (U_j \times I_\kappa) \times (I_n \times U_\kappa) \times \prod_{j=0}^{\kappa-1} (U_j \times I_\kappa), \tag{4.10}$$

where  $U_\kappa$  represents the function  $f_\kappa$  such that

$$f_\kappa = \prod_{j=0}^{\kappa-1} f_j(i), \tag{4.11}$$

and  $\prod_j^\kappa$  represents the addition modulo 2 for all  $\kappa$  functions, and eq. 4.8 can be rewritten as follows:

$$|\varphi_5\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |0\rangle^{\otimes \kappa} \otimes |f_\kappa(i)\rangle. \quad (4.12)$$

The main reason behind applying each oracle for the second time on its target qubit at each call of  $U_h$ , is that the solutions of that specific oracle are entangled with its target qubit. Discarding that qubit at the stage of amplifying the common solutions will drastically affect the desired outcome of the algorithm. So to get rid of this entanglement, applying each oracle on its respective target qubit is necessary to remove such correlation and maintain a valid result.

## 5 The Proposed Algorithm

In this section, we will propose the algorithm to find the common solutions  $M$  such that  $1 \leq M \leq N$ , among  $\kappa$  oracles, based on Younes *et al.* amplitude amplification algorithm. An illustration of the circuit is shown in fig. 4.

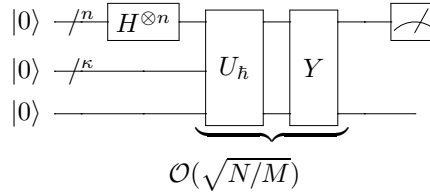


Figure 4: Quantum circuit for the proposed algorithm.

The algorithm is carried quantum mechanically as follows:

- 
- 1: Prepare the oracle  $U_h$ .
  - 2: Set the quantum register to  $|0\rangle^{\otimes n}$  and the extra  $\kappa + 1$  qubits to  $|0\rangle$ .
  - 3: Apply the Hadamard gates to the first  $n$  qubits to create the uniform superposition  $\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes |0\rangle^{\otimes \kappa+1}$ .
  - 4: Iterate over the following  $\frac{\pi}{2\sqrt{2}} \sqrt{\frac{N}{M}}$  steps:
    1. Apply the function  $U_h$ .
    2. Apply the diffusion operator  $Y$ .
  - 5: Measure the output
- 

## 6 Analysis and Comparison

In 2012, Tulsi proposed an algorithm [6] that given 2 oracles that can identify the elements of two sets, the goal is to find a common element between those



2 sets. The success of finding that single element is further enhanced using a variation Tulsı introduced of Grover’s amplitude amplification algorithm, with some restrictive conditions.

**Single Common Solution Amplification** In the case of a single common solution between  $\kappa = 2$  oracles, Tulsı’s algorithm is found to be optimal with restrictions, and requires  $\mathcal{O}(\sqrt{N})$  oracle calls. The proposed algorithm requires the same oracle calls  $\mathcal{O}(\sqrt{N})$  but with no restrictive conditions. In the case of single common solution when  $\kappa > 2$  oracles, the proposed algorithm is found to require  $\mathcal{O}(\sqrt{N})$  oracle calls.

**Multiple Common Solutions Amplification** In the case of multiple common solutions between  $\kappa = 2$  oracles, the expected function calls of the proposed algorithm is  $\mathcal{O}(\sqrt{N/M})$ , when  $M$  is  $1 \leq M \leq N$ . However, Tulsı’s algorithm can be used to cover the case of multiple solutions when  $\kappa = 2$ , but the problem becomes exponentially harder when  $M > 3N/4$  [5, 8]. In the case of multiple common solutions between  $\kappa > 2$  oracles, this case is not covered by Tulsı [6], however, the proposed algorithm requires  $\mathcal{O}(\sqrt{N/M})$  oracle calls.

## 7 Conclusion

In this literature, we proposed an algorithm to find the common entries  $M$  between  $\kappa$  sets, given the number of common entries. It was shown that the given oracles can be used to construct another oracle that exhibits the behavior of finding only the common entries between those oracles using two copies of each given oracle. The constructed oracle is used to mark the common entries with entanglement, then an amplitude amplification algorithm is applied to increase the success probability of finding the common entries.

## References

- [1] D. Deutsch and R. Jozsa, "Rapid Solution of Problems by Quantum Computation", *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 439, no. 1907, pp. 553-558, 1992. doi: 10.1098/rspa.1992.0167
- [2] H. Buhrman, R. Cleve, A. Wigderson, "Quantum vs. Classical Communication and Computation", *Proc. 30th ACM Symposium on Theory of Computing (STOC)*, 1998, 63-68. doi: 10.1145/276698.276713
- [3] L. Grover, "An Improved Quantum Scheduling Algorithm". Available: arXiv:quant-ph/0202033
- [4] C. Pang, R. Zhou, C. Ding and B. Hu, "Quantum search algorithm for set operation", *Quantum Information Processing*, vol. 12, no. 1, pp. 481-492, 2012. doi: 10.1007/s11128-012-0385-8

- [5] M. Boyer, G. Brassard, P. Høyer and A. Tapp, "Tight Bounds on Quantum Searching," *Fortschritte der Physik*, vol. 46, no. 4-5, pp. 493-505, 1998.
- [6] A. Tulsi, "Optimal quantum searching to find a common element of two sets". Available: [arXiv:1210.4648](https://arxiv.org/abs/1210.4648) [quant-ph]
- [7] L. Grover, "A fast quantum mechanical algorithm for database search," in *STOC 96 Proc. of the 28th annual ACM Syem. on Theory of computing*, 1996, pp. 212-219. doi: 10.1103/PhysRevLett.79.325
- [8] A. Younes, "Strength and weakness in Grover's quantum search algorithm". Available: [arXiv:0811.4481v1](https://arxiv.org/abs/0811.4481v1)
- [9] G. Brassard, P. Høyer, and A. Tapp, Quantum counting. Available: [arXiv:quant-ph/9805082](https://arxiv.org/abs/quant-ph/9805082)
- [10] A. Younes, J. Rowe and J. Miller, "Enhanced quantum searching via entanglement and partial diffusion," *Physica D: Nonlinear Phenomena*, vol. 237, no. 8, pp. 1074-1078, 2008. doi: 10.1016/j.physd.2007.12.005